Лекция 4

Тема. Объектно-ориентированного подхода к проектированию

План лекции

1. Основные положения объектного подхода.
2. Основные концепции объектно-ориентированного анализа, проектирования и программирования.
3. Объектно-ориентированный анализ и проектирование
4. Пример объектно-ориентированного анализа и проектирования

Основные положения объектного подхода

Методы структурного проектирования имели своей целью упростить процесс разработки сложных систем на основе алгоритмического подхода. Методы объектно-ориентированного проектирования созданы в свою очередь для помощи разработчикам, использующим мощные выразительные средства объектного и объектно-ориентированного программирования, основанного на описании классов и объектов. В принципах объектного подхода нашли свое отражение и множество других факторов, кроме объектно-ориентированного программирования (ООР). Объектный подход принципиально отличается от тех подходов, которые связаны с более традиционными методами структурного анализа, проектирования и программирования. Это не означает, что объектный подход требует отказа от всех ранее найденных и испытанных методов и приемов, напротив, новые элементы всегда основываются на предшествующем опыте. Объектный подход создает множество существенных удобств, которые при других условиях не могут быть обеспечены. Наиболее важным является то, что объектный подход позволяет создавать системы, которые воплощают атрибуты хорошо структурированных сложных систем. Кроме того, можно назвать несколько преимуществ, которые связаны с использованием объектного подхода.

Во-первых, объектный подход позволяет в полной мере использовать выразительные возможности объектных и объектно-ориентированных языков программирования. Существенно повысить эффективность и качество кода можно просто за счет использования C++ в качестве «улучшенного C» с элементами абстракции данных. Однако гораздо более значительным достижением является использование иерархии классов в процессе проектирования.

Во-вторых, использование объектного подхода существенно повышает качество разработки в целом и ее фрагментов. Объектно-ориентированные системы часто получаются более компактными, чем не объектно-ориентированные эквиваленты. А это означает не только уменьшение объема кода программ, но и удешевление проекта, и большее удобство в планировании разработок.

В-третьих, использование объектного подхода приводит к построению систем на основе стабильных промежуточных описаний, что упрощает процесс внесения изменений. Это дает системе возможность развиваться постепенно и не приводит к полной ее переработке в случае существенных изменений исходных требований.

Объектный подход уменьшает риск в разработке сложных систем, прежде всего потому, что процесс интеграции растягивается во времени жизненного цикла системы. Объектный подход состоит из ряда хорошо продуманных этапов проектирования, что также уменьшает степень риска и повышает уверенность в правильности принимаемых решений.

Основные концепции объектно-ориентированного анализа, проектирования и программирования

В основе объектно-ориентированного проектирования (OOD) лежит объектный подход. Основными принципами являются: абстрагирование, ограничение доступа, модульность, иерархичность, типизация, параллелизм и устойчивость. Эти принципы не новы, однако, именно в объектном подходе они объединены для решения общей задачи.

Объектно-ориентированное проектирование принципиально отличается от традиционных подходов структурного проектирования, так как подразумевает иной подход к процессу декомпозиции, а получаемый программный продукт по архитектуре в значительной степени выходит за рамки традиционных представлений. Отличия обусловлены тем, что структурное проектирование основано на структурном программировании, тогда как в основе объектно-ориентированного проектирования лежит методология объектно-ориентированного программирования (ООР).

ООР, OOD и ООА

Итак, объект - осязаемая сущность, которая четко проявляет свое поведение. А главным в понятии объекта является все же объединение идей абстрагирования данных и алгоритмов. Джонс уточняет это понятие следующим образом: «В объектном подходе акцент переносится на конкретные характеристики физической и абстрактной системы, являющейся предметом программного моделирования... Объекты обладают целостностью, которую не следует нарушать. Объект может только менять состояние, поведение, управляться или становиться в определенное отношение к другим объектам. Объект обладает неизменными качествами, но может изменять свое состояние. Например, подъемник характеризуется тем, что может двигаться вверх и вниз, оставаясь в пределах своих направляющих ... Любая модель должна учитывать эти свойства подъемника, так как они составляют его назначение».

Объектно-ориентированные программирование — это методология программирования, которая основана на представлении программы в виде совокупности объектов, каждый из которых является реализацией определенного *класса, а классы образуют иерархию на принципах наследуемости.*

В данном определении можно выделить три части: 1) ООР использует в качестве элементов конструкции объекты, а не алгоритмы; 2) каждый объект является реализацией какого-либо определенного класса; 3) классы организованы иерархически. Программа будет объектно-ориентированной только при соблюдении всех трех, указанных требований. В частности, программирование, не основанное на иерархических отношениях, не относится к ООР, а называется программированием на основе абстрактных типов данных.

Поддержка наследуемости в таких языках означает возможность установления отношений вида «разновидность» между типами, например красная роза - разновидность цветов, а цветок - разновидность растений. Языки, не имеющие таких механизмов, нельзя отнести к объектно-ориентированным.

Методы программирования, прежде всего подразумевают правильное и эффективное использование механизмов языков программирования. Методы проектирования, напротив, основное внимание направляют на правильное и эффективное структурирование сложных систем.

Объектно-ориентированное проектирование — это методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления как логической и физической, так статической и динамической моделей проектируемой системы.

В данном определении содержатся две важные части: 1) OOD ведет к объектно-ориентированной декомпозиции; 2) используется многообразие приемов представления

моделей, отражающих логическую (структуры классов и объектов) и физическую (архитектура моделей и процессов) структуру системы.

Именно поддержка объектно-ориентированной декомпозиции отличает OOD от структурного проектирования; в первом случае логическая структура системы отражается абстракциями в виде классов и объектов, во втором в виде алгоритмического обозначения методов, связанных с объектно-ориентированной декомпозицией.

Объектно-ориентированный анализ (ООА) направлен на создание моделей, более близких к реальности, с использованием объектно-ориентированного подхода; это методология, при которой требования формируются на основе понятий классов и объектов, составляющих словарь предметной области.

На результатах ООА формируются модели, на которых основывается OOD; OOD в свою очередь создает основу для окончательной реализации системы с использованием методологии ООР.

Объектно-ориентированный анализ и проектирование

Процесс объектно-ориентированного анализа и проектирования (как и разработка программной системы в целом) тесно связан с подготовительным этапом или анализом требований, включающим в себя описание прецедентов.

Наиболее важным моментом *ООА/П* является квалифицированное распределение обязанностей между программными компонентами.

На каком основании это можно утверждать? Дело в том, что это единственный вид деятельности, без которого невозможно обойтись. Кроме того, он оказывает определяющее влияние на робастность, масштабируемость, расширяемость и возможность повторного использования компонентов.

Конечно же, в процессе *ООА/П* решаются и другие важные вопросы, однако определению обязанностей уделяется особое внимание, поскольку это самая важная и жизненно важная задача. Поэтому при описании этапа проектирования основное внимание уделяется определению обязанностей.

Этап *анализа* состоит в исследовании системных требований и проблем, а не в поисках путей ее решения. Например, при разработке новой информационной системы для компьютерной библиотеки необходимо описать экономические процессы, связанные с ее использованием.

Анализ - это достаточно широкое понятие. Его содержание более точно отражают термины *анализ требований*) (т.е. исследование требований к системе) и *объектный анализ* (исследование объектов предметной области).

В процессе *проектирования* основное внимание уделяется концептуальному решению, обеспечивающему выполнение основных требований, но не вопросам его реализации. Например, на этапе проектирования описываются программные объекты или схема базы данных. Конечно же, проектные решения могут быть реализованы в программном коде.

Как и понятие анализа, этот термин тоже стоит уточнить и говорить об *объектном проектировании* или *проектировании базы данных*.

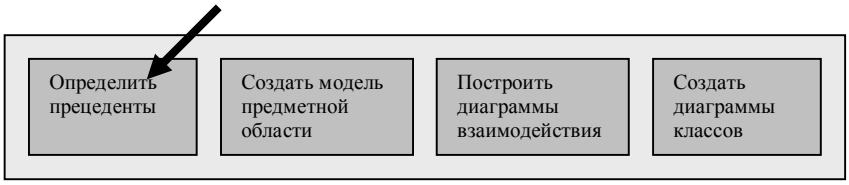
В процессе объектно-ориентированного анализа основное внимание уделяется определению и описанию объектов (или понятий) в терминах предметной области. В процессе объектно-ориентированного проектирования определяются программные объекты и способы их взаимодействия с целью выполнения системных требований. И, наконец, на этапе реализации или объектно-ориентированного программирования обеспечивается реализация разработанных компонентов на языке *Java*.

Пример объектно-ориентированного анализа и проектирования

Прежде чем вдаваться в детали анализа требований и ООА/П, рассмотрим основные действия, выполняемые в процессе разработки программной системы, и создаваемые при этом диаграммы. В качестве примера возьмем игру в кости, в которой игрок бросает два кубика. Если сумма очков равна семи, участник считается победителем, в противном случае — проигравшим.

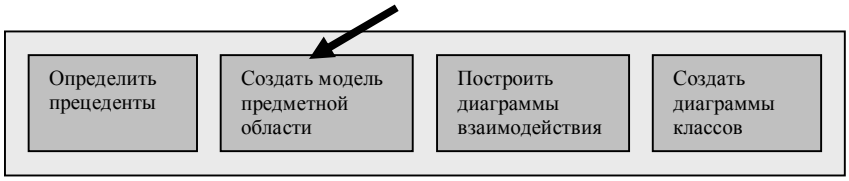
Анализ требований может включать описание процессов предметной области, представленное в форме прецедентов.

Прецеденты не являются артефактами объектно-ориентированно го анализа — это просто повествовательные истории. Однако они составляют важный этап анализа требований и унифицированного процесса в целом.



Например, в процессе игры в кости может присутствовать прецедент Игра в кости со следующим кратким описанием: *Игра в кости.* Игрок берет и бросает кости. Если сумма очков составляет 7, игрок считается победителем, в противном случае — проигравшим.

Объектно-ориентированный анализ связан с описанием предметной области с точки зрения классификации объектов. Декомпозиция предметной области задачи состоит в идентификации понятий, атрибутов и ассоциаций из предметной области, имеющих важное значение для решения задачи. Результат анализа выражается в *модели предметной области*, которая иллюстрируется с помощью набора диаграмм с изображенными на них понятиями или объектами предметной области



Например, модель предметной области игры в кости показана на рис. 2.1 Она иллюстрирует понятия *Player* (игрок), *Die* (кость) и *DiceGame* (игра в кости), а также их связи и атрибуты.

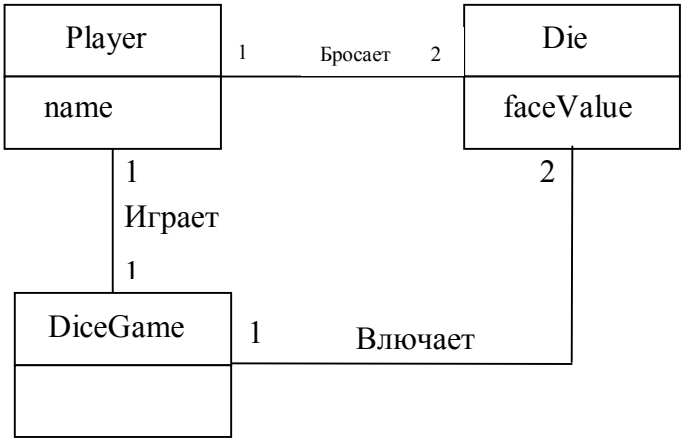
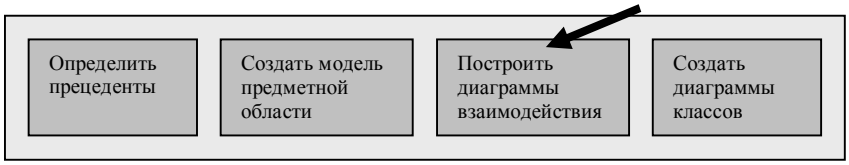


Рис.2.1. Модель предметной области игры в кости

Модель предметной области — это не описание программных компонентов. Это представление понятий, выраженных в терминах предметной области задачи.

Диаграммы взаимодействий

Объектно-ориентированное проектирование связано с определением программных компонентов и способов их взаимодействия. Для иллюстрации взаимосвязей между объектами используется диаграмма взаимодействий (*collaboration diagram*), отображающая потоки передачи сообщений программными объектами и вызовы методов.



Например, рассмотрим задачу программного моделирования игры в кости. Диаграмма взаимодействий объектов этой системы иллюстрирует важные этапы игры передачу сообщений между экземплярами классов *DiceGame* и *Die* (рис. 2.2.).

Обратите внимание на следующую особенность. Хотя в реальном мире кости подбрасывает игрок, в программной системе эту функцию выполняет объект *DiceGame* (т.е. передает сообщения объектам *Die*).

Дело в том, что программные объекты в некотором смысле соответствуют объектам реального мира, но не являются их точными моделями или копиями.

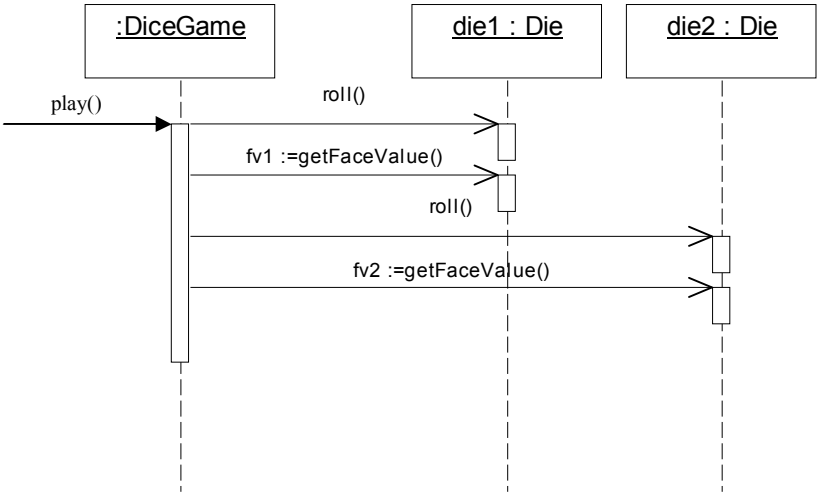
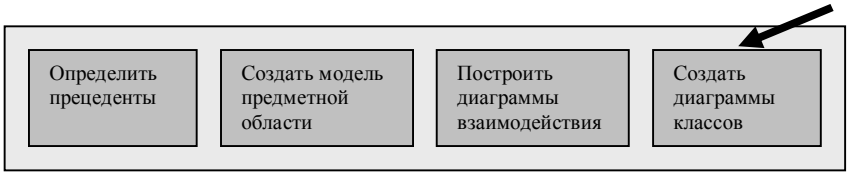


Рис.2.2.Диаграмма взаимодействий, иллюстрирующая передачу сообщений между программными объектами.

Диаграммы классов

Помимо *динамического представления* взаимосвязи объектов, отображаемой на диаграмме взаимодействий, очень полезно строить *статическое представление* системы в виде *диаграммы классов проектирования* (*design class diagram*). На такой диаграмме отображаются атрибуты и методы классов.



Например, анализ диаграммы взаимодействия игры в кости приводит к диаграмме классов, показанной на рис. 2.3. Поскольку сообщение *play* передается объекту *DiceGame*, класс *DiceGame* должен содержать метод *play*, а класс *Die* — методы *roll* и *getFaceValue*.

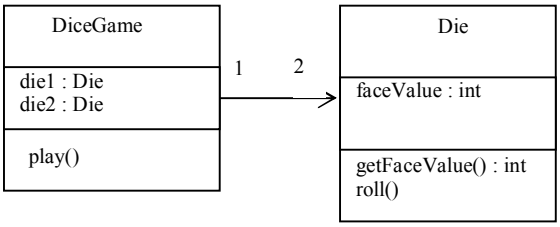


Рис.2.3. Диаграмма классов программных компонентов

В отличие от модели предметной области, эта диаграмма не иллюстрирует понятия реального мира, а описывает программные классы.

Подытоживая вышесказанное, объектно-ориентированный анализ и проектирование можно определить как «выявление правильных действий (анализ) и обеспечение их правильности (проектирование)».

Вопросы для самопроверки.

1. Какие преимущества объектного подхода ?
2. Что такое объектно-ориентированное программирование?
3. Что такое объектно-ориентированное проектирование?
4. Что такое объектно-ориентированный анализ?
5. Значение декомпозиции предметной области задачи?